Electrical Flows, Laplacian Matrices, and New Approaches to the Maximum Flow Problem

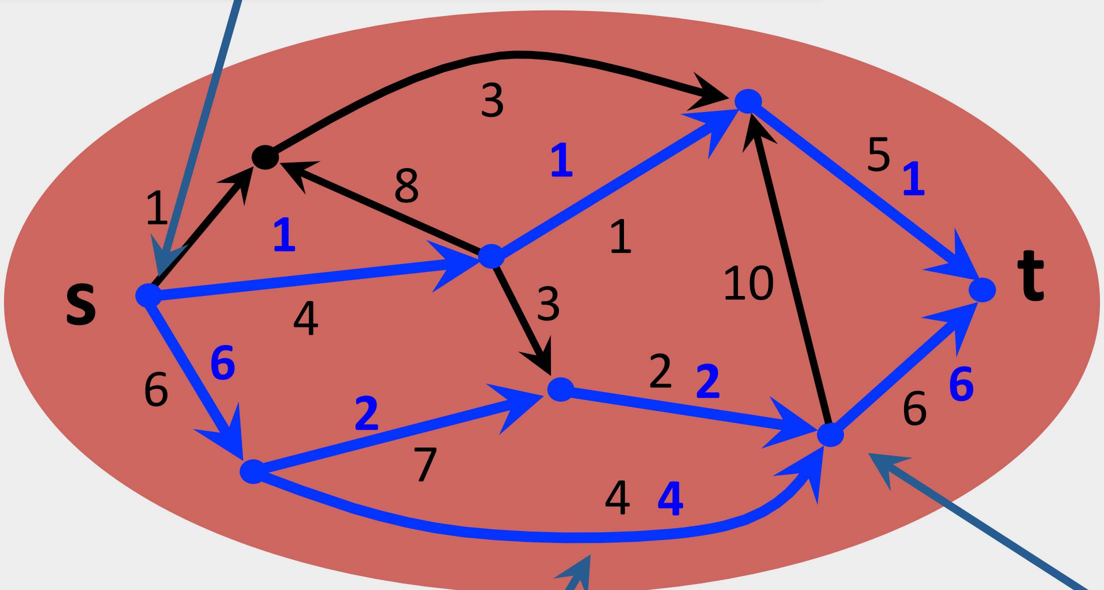
Aleksander Mądry

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Maximum flow problem

Input: Directed graph **G**,
integer **capacities u_e**,
source s and **sink t**

value = net flow out of **s**



Max flow value
F*=10

no overflow on arcs:
$0 \le f(e) \le u(e)$

no leaks at all **v≠s,t**

Task: Find a **feasible s-t flow** of **max value**

Breaking the $O(n^{3/2})$ barrier

Undirected graphs and approx. answers ($O(n^{3/2})$ barrier still holds here)

[M '10]: **Crude approx. of** max flow **value** in **close to linear** time

[CKMST '11]: **(1-ε)-approx.** to max flow in $\tilde{O}(n^{4/3}\varepsilon^{-3})$ time

[LSR '13, S '13, KLOS '14]: **(1-ε)-approx.** in **close to linear** time

But: What about the **directed** and **exact** setting?

Yesterday

[M '13]: Exact $\tilde{O}(n^{10/7})=\tilde{O}(n^{1.43})$-time alg.

(n = # of vertices, $\tilde{O}()$ hides polylog factors)

Breaking the $O(n^{3/2})$ barrier

Undirected graphs and approx. answers ($O(n^{3/2})$ barrier still holds here)

[M '10]: **Crude approx. of** max flow **value** in **close to linear** time

[CKMST '11]: **(1-ε)-approx.** to max flow in $\tilde{O}(n^{4/3}\varepsilon^{-3})$ time

[LSR '13, S '13, KLOS '14]: **(1-ε)-approx.** in **close to linear** time

But: What about the **directed** and **exact** setting?

[M '13]: Exact $\tilde{O}(n^{10/7})=\tilde{O}(n^{1.43})$-time alg.

Today

(n = # of vertices, $\tilde{O}()$ hides polylog factors)

From electrical flows to **exact directed** max flow

From now on: All capacities are **1**, **$m=O(n)$**
and the value **F^*** of max flow is known

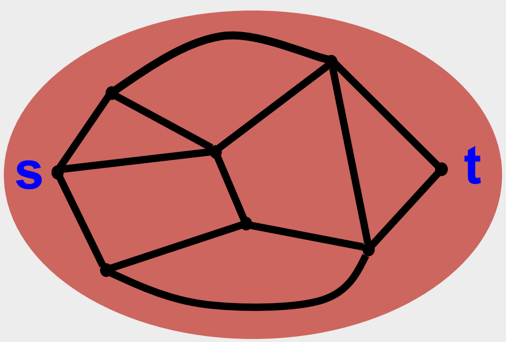
Why the progress on **approx. undirected** max flow does not apply to the **exact directed** case?

Tempting answer: Directed graphs are just different (for one, electrical flow is an undirected notion)

But: exact directed max flow reduces to **exact**

We need a more powerful intermediary

So, it is all about getting n



Key obstacle: Gradient descent methods (like MWU) are inherently unable to deliver good enough accuracy

(Path-following) Interior-point method (IPM)

[Dikin '67, Karmarkar '84, Renegar '88,…]

A powerful framework for solving general LPs (and more)

LP: $\min c^T x$

s.t. $Ax = b$

$x \geq 0$

Idea: Take care of "hard" constraints by adding a "barrier" to the objective

"**easy**" constraints
(use projection)

"**hard**" constraints

(Path-following) Interior-point method (IPM)

[Dikin '67, Karmarkar '84, Renegar '88,…]

A powerful framework for solving general LPs (and more)

LP(μ): min $c^T x - \mu \sum_i \log x_i$

s.t. $Ax = b$

$\cancel{x \geq 0}$

Idea: Take care of "hard" constraints by adding a "barrier" to the objective

Observe: The barrier term enforces **$x \geq 0$** implicitly

Furthermore: for large **μ**, **LP(μ)** is easy to solve and

LP(μ) → original LP, as $\mu \to 0^+$

Path-following routine:

→ Start with (near-)optimal solution to **LP(μ)** for large **$\mu>0$**

→ Gradually reduce **μ** while maintaining the (near-)optimal solution to current **LP(μ)**

(Path-following) Interior-point method (IPM)

[Dikin '67, Karmarkar '84, Renegar '88,…]

A powerful framework for solving general LPs (and more)

LP(μ): min $c^T x$ - μ Σ_i log x_i
s.t. Ax = b
~~**x ≥ 0**~~

Idea: Take care of "hard" constraints by adding a "barrier" to the objective

Observe: The barrier term enforces $x \geq 0$ implicitly

Based on **second-order approx.**

$$f(x+y) \approx f(x) + y^T \nabla f(x) + y^T H_f(x) y$$

+ projection on ker(A)

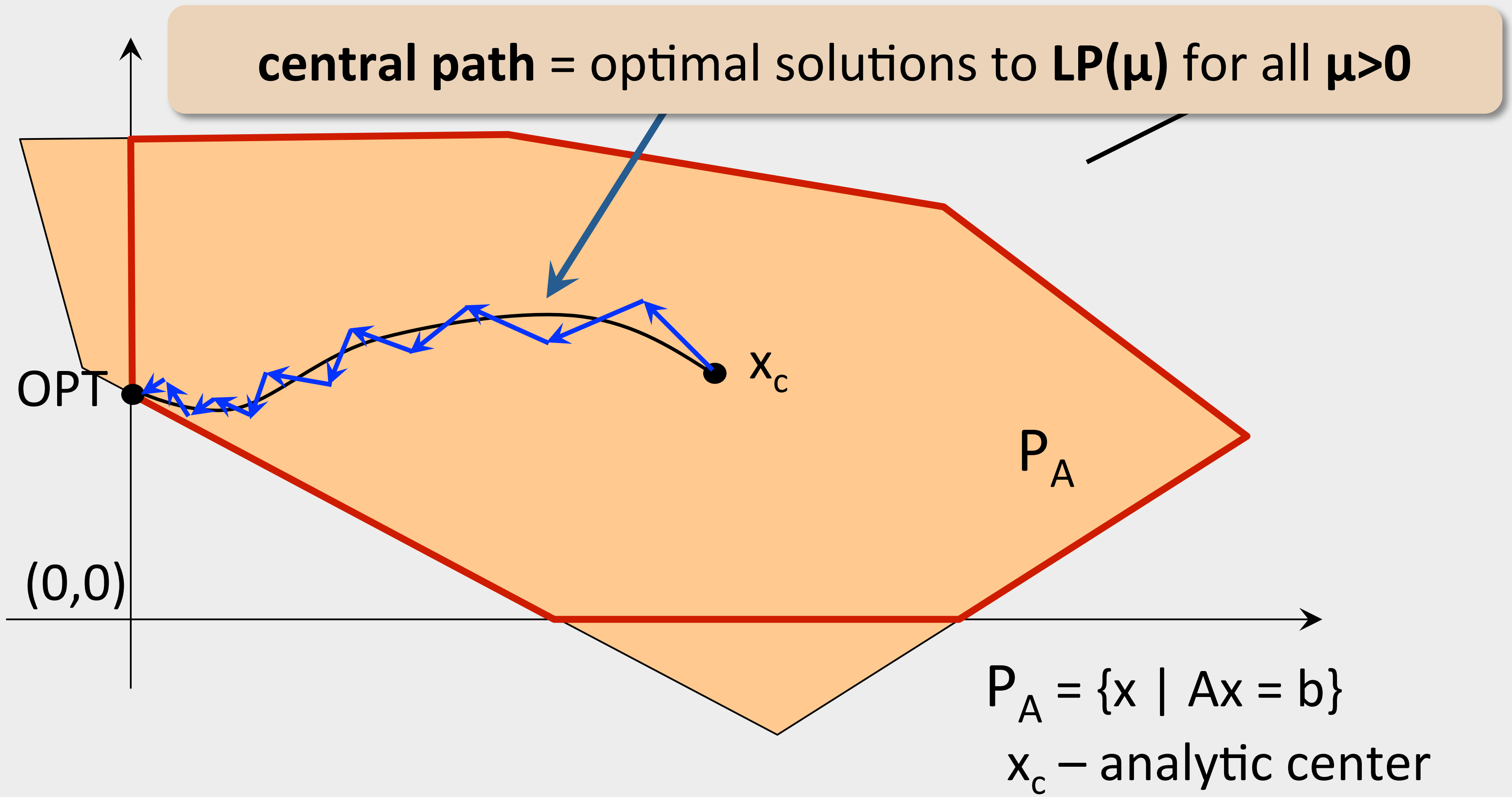
Path-following routine:

→ Maintain (near-)optimal solutio

→ Repeat:

Set **μ'=(1-δ)μ** and use **Newton's method** to compute from **x** (near-)optimal solution to **LP(μ')**

Key point: Choosing step size **δ** sufficiently small ensures **x** is close to optimum for **LP(μ')** → Newton's method convergence **very** rapid



Path-following routine:

→ Start with (near-)optimal solution to **LP(μ)** for large **μ>0**

→ Gradually reduce **μ** (via **Newton's method**) while maintaining the (near-)optimal solution to current **LP(μ)**

Can we use IPM to get a faster max flow alg.?

Conventional wisdom: This will be too slow!

→ Each **Newton's step** = solving a linear system $\mathbf{O(n^{\omega})=O(n^{2.373})}$ time (prohibitive!)

But: When solving **flow problems** – only $\mathbf{\tilde{O}(m)}$ time **[DS '08]**

Fundamental question: What is the number of iterations?

[Renegar '88]: $\mathbf{O(m^{1/2} \log \varepsilon^{-1})}$

Unfortunately: This gives only an $\mathbf{\tilde{O}(m^{3/2})}$-time algorithm

Improve the $\mathbf{O(m^{1/2})}$ bound?
Although believed to be **very** suboptimal,
its improvement is a major challenge

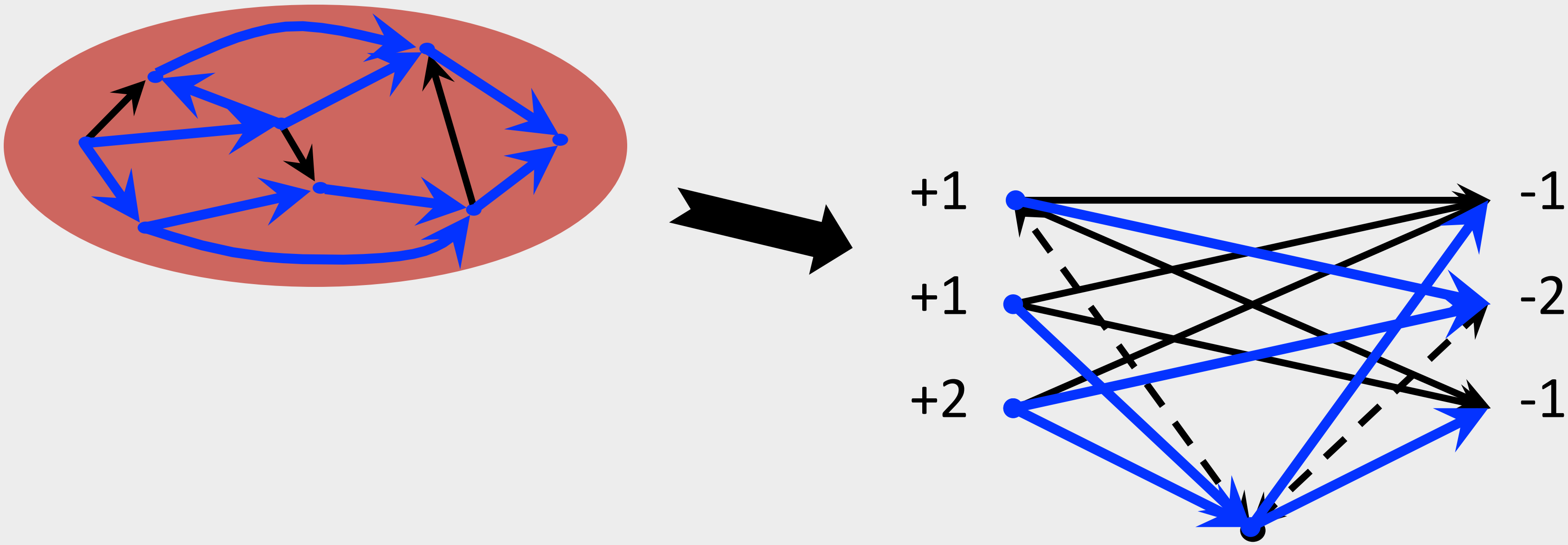
?

The Max Flow algorithm

(Self-contained, but can be seen as a variation on IPM)

From Max Flow to Min-cost Flow

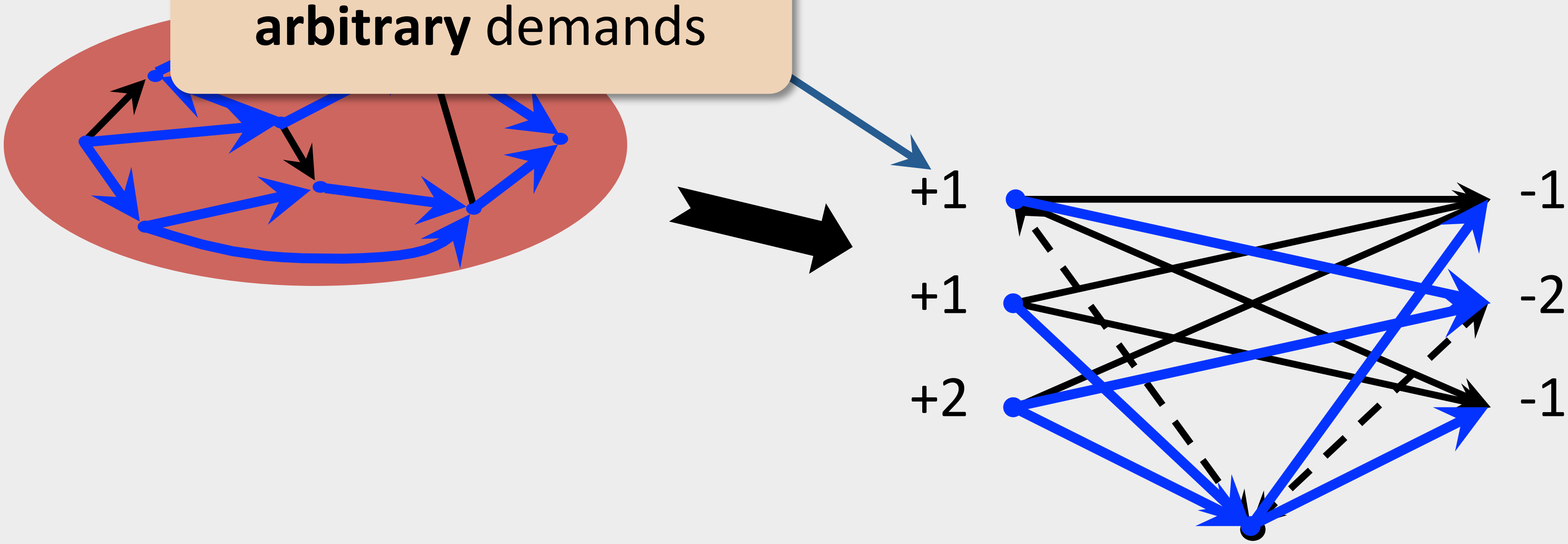
Reduce **max flow** to **uncapacitated min-cost σ-flow problem**



From Max Flow to Min-cost Flow

Reduce **max flow** to **uncapacitated min-cost σ-flow problem**

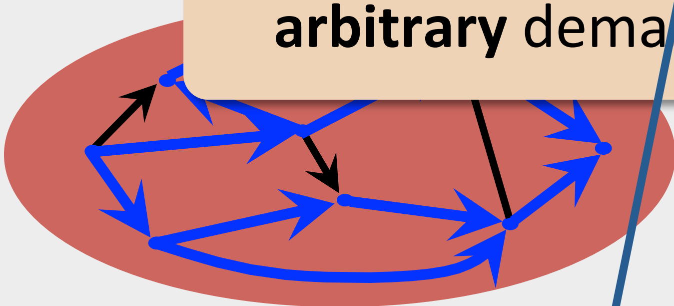
arbitrary demands

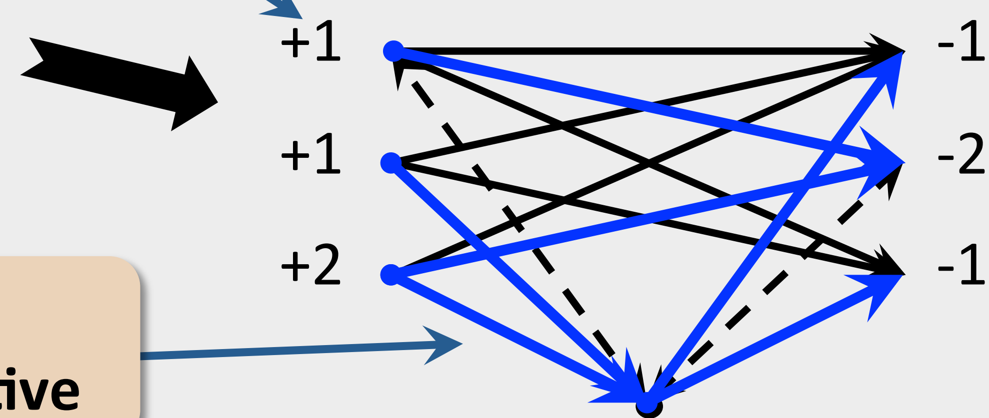


From Max Flow to Min-cost Flow

Reduce **max flow** to **uncapacitated** **min-cost** **σ-flow problem**

arbitrary demands

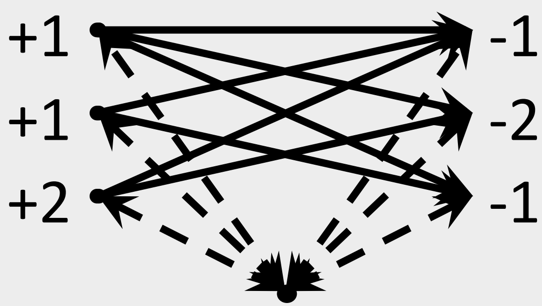




no capacity constraints
but flow has to be **non-negative**

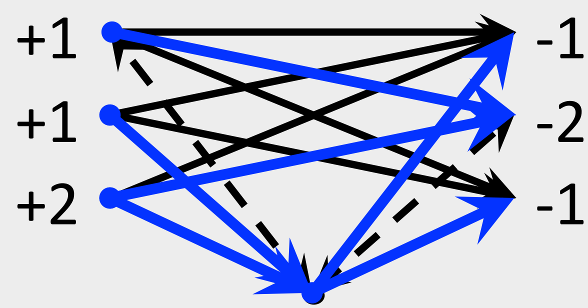
Result: Feasibility → Optimization
+ special structure

Solving Min-Cost Max Flow Instance



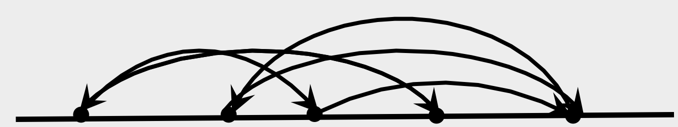
Our approach is **primal-dual**

→ **Primal** solution: **σ-flow f**
(**feasibility:** all $\mathbf{f_e}$ are **≥0**)

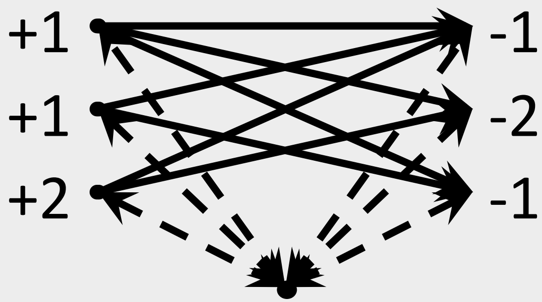


→ **Dual** solution: **embedding y into real line**
(**feasibility:** all **slacks** $\mathbf{s_e}$ are **≥0**)

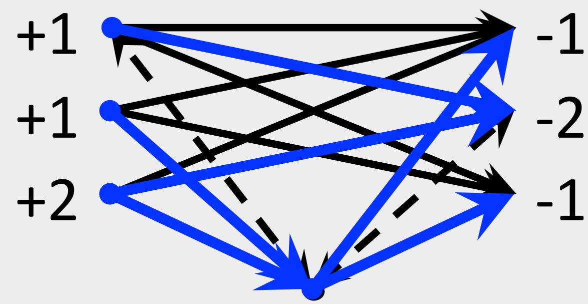
"No arc is too stretched"



Solving Min-Cost Max Flow Instance



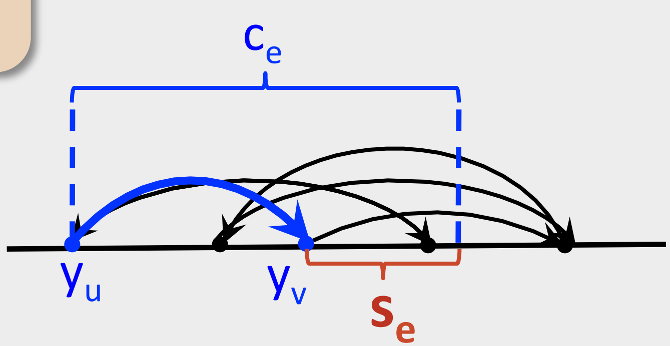
Our approach is **primal-dual**



→ **Primal** solution: **σ-flow f**
(**feasibility:** all $\mathbf{f_e}$ are **≥0**)

→ **Dual** solution: **embedding y into real line**
(**feasibility:** all **slacks** $\mathbf{s_e}$ are **≥0**)

"No arc is too stretched"



Solving Min-Cost Max Flow Instance

Our Goal:

Get **(f,y)** with small **duality gap** $\Sigma_e f_e s_e$

Our Approach: Iteratively improve maintained solution while enforcing an **additional** constraint

Centrality:

$f_e s_e \approx \mu$, for all **e**

(with **μ** being progressively smaller)

"Make all arcs have similar contribution to the duality gap"

(Maintaining **centrality** = following the **central path**)

Taking an Improvement Step

So far, our approach is fairly standard

Crucial Question:
How to improve the quality of maintained solution?

Key Ingredient:
Use electrical flows

Taking an Improvement Step

Let **(f,y)** be a (centered) primal-dual solution

Key step: Compute **electrical σ-flow f^+** with $r_e := s_e / f_e$

Primal improvement: Set $f' := (1-\delta)f + \delta f^+$

Dual improvement: Use **voltages φ** inducing f^+ (via Ohm's Law)
Set $y' := y + \delta(1-\delta)^{-1} \varphi$

Can show: When terms **quadratic** in **δ** are ignored

$$f_e' \, s_e' \approx (1-\delta)\,\mu = \mu'$$
for each **e**

(i.e., **duality gap** decreases by **(1-δ)** and **centrality** is preserved)

How big **δ** can we take to have this approx. hold?

Lowerbounding δ

Can show:
$\boldsymbol{\delta}^{-1}$ is bounded by $\mathbf{O(|\rho|_4)}$
where $\boldsymbol{\rho_e := |f_e^+|/f_e}$

$\mathbf{|\rho|_4}$ measures
how different $\mathbf{f^+}$ and $\mathbf{f}$ are

How to bound $\mathbf{|\rho|_4}$?

Idea: Bound $\mathbf{|\rho|_2 \geq |\rho|_4}$ instead

Lowerbounding δ

Can show:
$\boldsymbol{\delta}^{-1}$ is bounded by $\mathbf{O}(|\boldsymbol{\rho}|_4)$
where $\boldsymbol{\rho}_e := |f_e^+|/f_e$

$|\boldsymbol{\rho}|_4$ measures
how different $\mathbf{f}^+$ and $\mathbf{f}$ are

How to bound $|\boldsymbol{\rho}|_2$? $(|\boldsymbol{\rho}|_2 \geq |\boldsymbol{\rho}|_4)$

Centrality: Tying $|\boldsymbol{\rho}|_2$ to $\mathbf{E}(\mathbf{f}^+)$

$f_e s_e \approx \mu \rightarrow r_e = s_e/f_e \approx \mu/(f_e)^2$

$\Downarrow$

$\mathbf{E}(\mathbf{f}^+) \approx \mu\,(|\boldsymbol{\rho}|_2)^2$

Lowerbounding δ

Can show:
$\boldsymbol{\delta}^{-1}$ is bounded by $\mathbf{O}(|\boldsymbol{\rho}|_4)$
where $\boldsymbol{\rho}_e := |f_e^+|/f_e$

$|\boldsymbol{\rho}|_4$ measures
how different $\mathbf{f}^+$ and $\mathbf{f}$ are

How to bound $|\boldsymbol{\rho}|_2$? $(|\boldsymbol{\rho}|_2 \geq |\boldsymbol{\rho}|_4)$

Centrality: Tying $|\boldsymbol{\rho}|_2$ to $\mathbf{E}(\mathbf{f}^+)$

$$f_e s_e \approx \mu \rightarrow r_e = s_e/f_e \approx \mu/(f_e)^2$$

$$\Downarrow$$

$$E(f^+) = \Sigma_e\, r_e\, (f_e^+)^2 \approx \Sigma_e\, \mu\, (f_e^+/f_e)^2 = \mu\, \Sigma_e\, (\rho_e)^2 = \mu\, (|\rho|_2)^2$$

So, we can focus on bounding $\mathbf{E}(\mathbf{f}^+)$

Lowerbounding δ

Can show:
$\boldsymbol{\delta}^{-1}$ is bounded by $\mathbf{O}(|\boldsymbol{\rho}|_4)$
where $\boldsymbol{\rho}_e := |f_e^+|/f_e$

$|\boldsymbol{\rho}|_4$ measures
how different $\mathbf{f}^+$ and $\mathbf{f}$ are

How to bound $|\boldsymbol{\rho}|_2$? $\quad (|\boldsymbol{\rho}|_2 \geq |\boldsymbol{\rho}|_4)$

How to bound $\mathbf{E}(\mathbf{f}^+)$? $\quad (\mathbf{E}(\mathbf{f}^+) \approx \mu\, (|\boldsymbol{\rho}|_2)^2)$

Idea: Use energy-bounding argument we used in the undirected case

Claim: $\mathbf{E}(\mathbf{f}^+) \leq \mu m$

Proof: Note that $\mathbf{E}(\mathbf{f}) = \Sigma_e\, r_e\, (f_e)^2 \approx \Sigma_e\, \mu\, (f_e/f_e)^2$

Lowerbounding δ

Can show:
$\boldsymbol{\delta^{-1}}$ is bounded by $\mathbf{O(|\rho|_4)}$
where $\boldsymbol{\rho_e := |f_e^+|/f_e}$

$|\rho|_4$ measures
how different $\mathbf{f^+}$ and $\mathbf{f}$ are

How to bound $|\rho|_2$? $\quad (|\rho|_2 \geq |\rho|_4)$

How to bound $\mathbf{E(f^+)}$? $\quad (E(f^+) \approx \mu\, (|\rho|_2)^2)$

Idea: Use energy-bounding argument we used in the undirected case

Claim: $\mathbf{E(f^+) \leq \mu m}$

Proof: Note that $\mathbf{E(f) = \Sigma_e\, r_e\, (f_e)^2 \approx \Sigma_e\, \mu\, (f_e/f_e)^2 = \mu\, \Sigma_e\, 1 = \mu m}$

Result: Bounding $\boldsymbol{\delta^{-1} \leq |\rho|_4 \leq |\rho|_2 \leq (E(f^+)/\mu)^{1/2} \leq m^{1/2}}$

$\mathbf{E(f^+) \leq E(f) \approx \mu m}$

This recovers the canonical $\mathbf{O(m^{1/2})}$-iterations bound for **general IPMs** and gives the $\mathbf{\tilde{O}(m^{3/2} \log U)}$ algorithm

Going beyond $\Omega(m^{1/2})$ barrier

Our reasoning before: $\delta^{-1} \leq |\rho|_4 \leq |\rho|_2 \leq m^{1/2}$

Essentially tight
in our framework

Going beyond $\Omega(m^{1/2})$ barrier

Our reasoning before: $\delta^{-1} \leq |\rho|_4 \leq |\rho|_2 \leq m^{1/2}$

When does $|\rho|_4 \approx |\rho|_2$?

This part we need to improve

Going beyond $\Omega(m^{1/2})$ barrier

Our reasoning before: $\delta^{-1} \leq |\rho|_4 \leq |\rho|_2 \leq m^{1/2}$

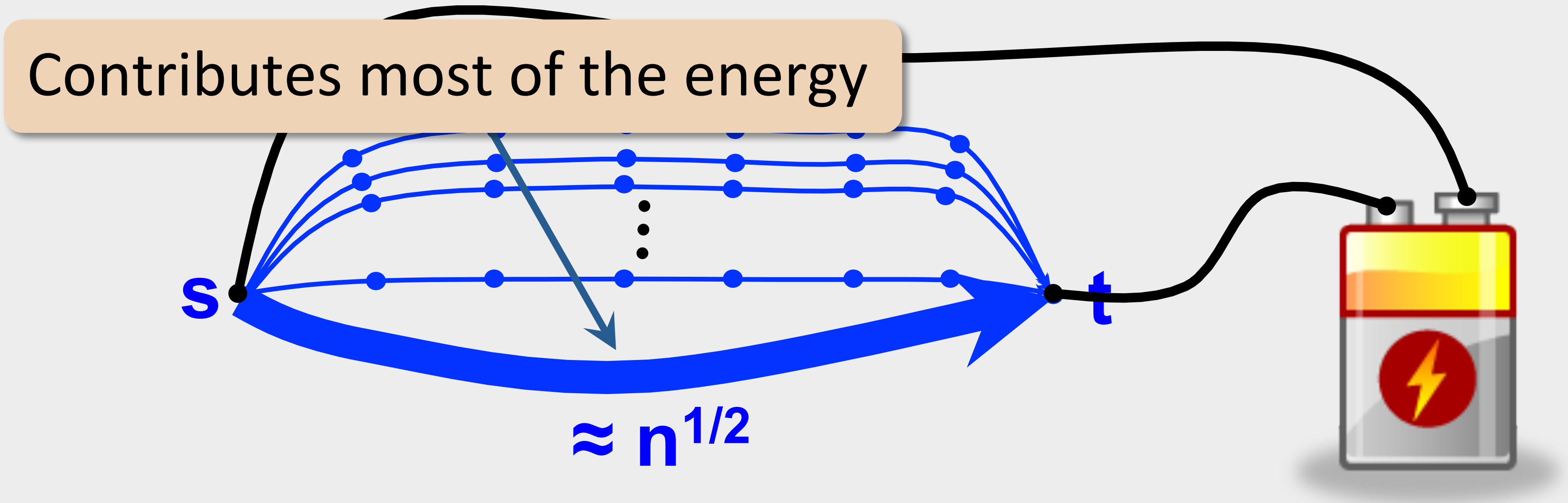
When does $|\rho|_4 \approx |\rho|_2$?

Answer: If most of the norm of ρ is focused on only a few coordinates

Translated to our setting: $|\rho|_4 \approx |\rho|_2$ if most of the energy of f^+ is contributed by only a few arcs

Can this happen?

Unfortunately, yes



Going beyond $\Omega(m^{1/2})$ barrier

Our reasoning before: $\delta^{-1} \leq |\rho|_4 \leq |\rho|_2 \leq m^{1/2}$

When does $|\rho|_4 \approx |\rho|_2$?

Answer: If most of the norm of $\boldsymbol{\rho}$ is focused on only a few coordinates

Translated to our setting: $|\rho|_4 \approx |\rho|_2$ if most of the energy of $\mathbf{f}^+$ is contributed by only a few arcs

Can this happen?

Unfortunately, yes

in principle, tight)

This is the **only** part where **unit-capacity** assumption is needed

ppen too often

Method: Very careful perturbation of the solution

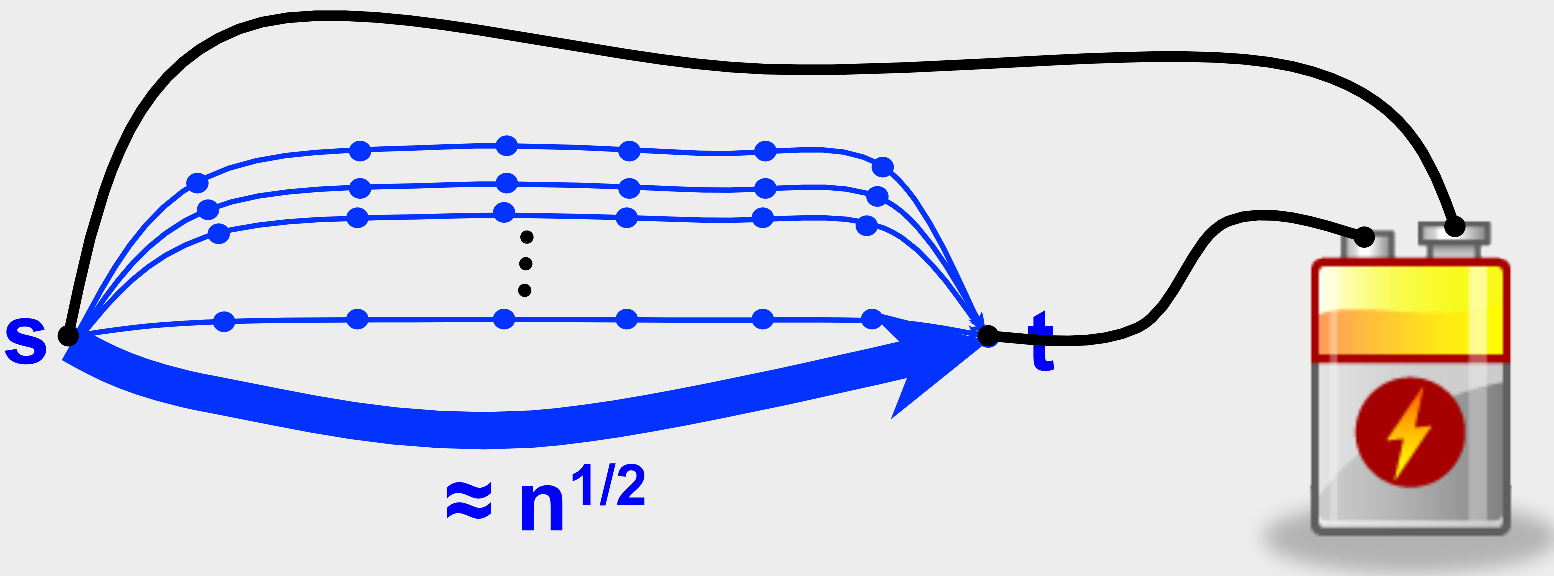
+ certain **preconditioning**

Going beyond $\Omega(m^{1/2})$ barrier

Problematic case: When most of the energy of $\mathbf{f}^+$ is contributed by only a few arcs

How can we ensure that this is not the case?

We already faced such problems in the undirected setting!



Going beyond $\Omega(m^{1/2})$ barrier

Problematic case: When most of the energy of $\mathbf{f^+}$ is contributed by only a few arcs

How can we ensure that this is not the case?

We already faced such problems in the undirected setting!

Our approach then: Keep removing high-energy edges

To show this works: Used the energy of the electrical flow as a potential function

- Energy **can only increase** and obeys global upper bound
- Each time removal happens → energy **increases by a lot**

Problems: In our framework, arc removal is **too drastic** and the energy of $\mathbf{f^+}$ is **highly non-monotone**

Going beyond $\Omega(m^{1/2})$ barrier

How to deal with these problems?

→ Enforce a **stronger** condition than just that $\|\rho\|_4$ is small (**"smoothness":** restrict energy contributions of arc subsets)

Key fact: f^+ smooth → energy does **not** change too much (so, energy becomes a good potential function again)

→ To enforce this, keep **stretching** the offending arcs (**stretch =** increase length by s_e - this doubles the resistance $r_e=s_e/f_e$)

As long as s_e is small for stretched arcs, the resulting perturbation of lengths can be corrected at the end

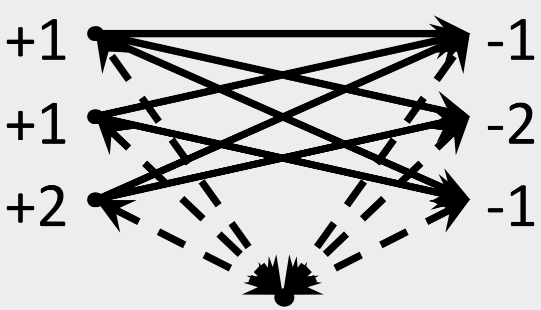
Remaining question: How to handle arcs with **large** s_e?

Going beyond $\Omega(m^{1/2})$ barrier

Observation: As $\mathbf{f_e s_e} \approx \boldsymbol{\mu}$, large $\mathbf{s_e}$ → small flow $\mathbf{f_e}$ and thus $\mathbf{r_e = s_e/f_e} \approx \boldsymbol{\mu}/\mathbf{f_e}^2$ is pretty large

→ **For such arcs:** contributing a lot of energy implies high <u>**effective**</u> resistance

Idea: Precondition **(f,y)** so as no arc has too high effect. resist.



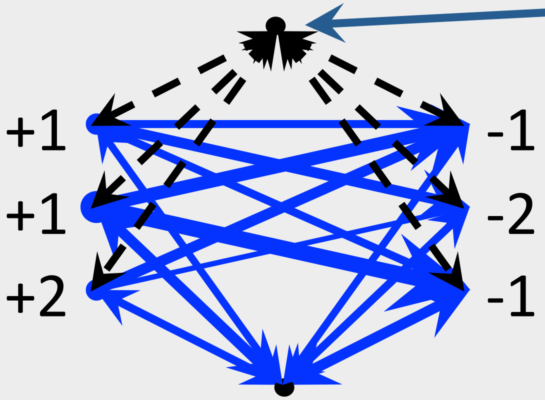
Going beyond $\Omega(m^{1/2})$ barrier

Observation: As $\mathbf{f_e s_e} \approx \boldsymbol{\mu}$, large $\mathbf{s_e}$ → small flow $\mathbf{f_e}$ and thus $\mathbf{r_e = s_e/f_e} \approx \boldsymbol{\mu}/\mathbf{f_e}^2$ is pretty large

→ **For such arcs:** contributing a lot of energy implies high **effective** resistance

Idea: Precondition **(f,y)** so as no arc has too high effect. resist.

Auxiliary star graph

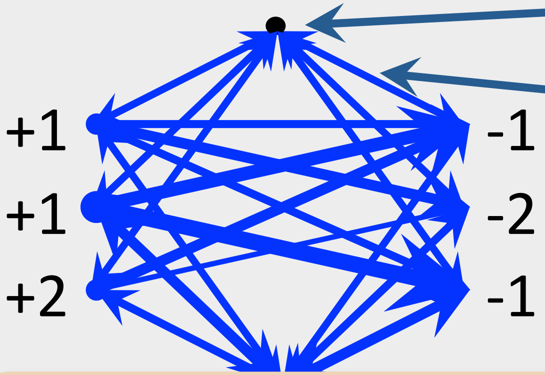


Going beyond $\Omega(m^{1/2})$ barrier

Observation: As $f_e s_e \approx \mu$, large $s_e \rightarrow$ small flow f_e and thus $r_e = s_e/f_e \approx \mu/f_e^2$ is pretty large

$\rightarrow$ **For such arcs:** contributing a lot of energy implies high **effective** resistance

Idea: Precondition **(f,y)** so as no arc has too high effect. resist.

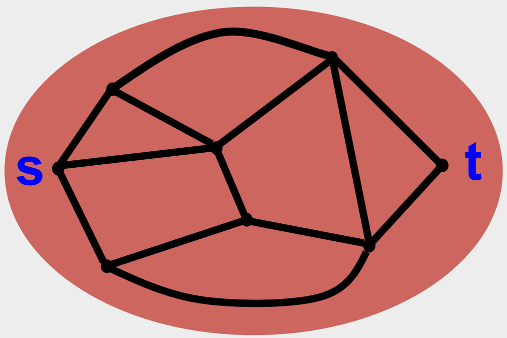


Can show: After doing that, no arc with large s_e

Putting these two techniques together + some work:
$\tilde{O}(m^{3/7})$**-iterations convergence follows**

Conclusions
and the Bigger Picture

Maximum Flows and Electrical Flows



Interior-point method

Elect. flows + IPMs → A powerful new approach to **max flow**

Can this lead to a **nearly-linear time** algorithm for the **exact directed** max flow?

We seem to have the "critical mass" of ideas

Elect. flows = next generation of "spectral" tools?

- Better "spectral" graph partitioning,
- Algorithmic grasp of random walks,
- …

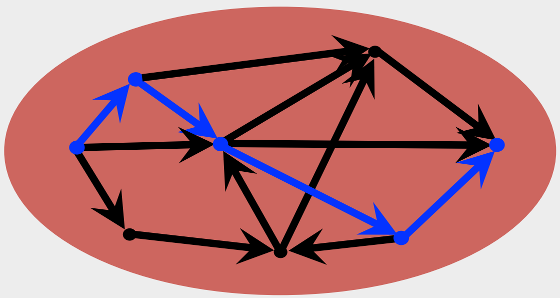
Grand challenge: Can we make algorithmic graph theory run in nearly-linear time?

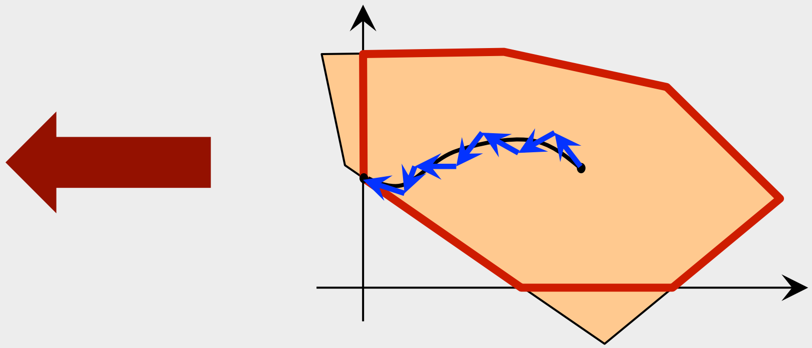
New "recipe": Fast alg. for **combinatorial** problems via **linear-algebraic** tools **+ continuous opt.** methods

How about applying this framework to other graph problems that "got stuck" at $\mathbf{O(n^{3/2})}$?

(min-cost flow, general matchings, negative-lengths shortest path…)

Second-order/IPM-like methods:
the next frontier for fast (graph) algorithms?





Max Flow and Interior-Point Methods

Contributing back: Max flow and electrical flows as a lens for analyzing general IPMs?

Our techniques can be lifted to the general LP setting

We can solve **any** LP within $\tilde{O}(m^{3/7}L)$ iterations
But: this involves **perturbing** of this LP

Some (seemingly) new elements of our approach:

- Better grasp of ℓ_2 vs. ℓ_4 interplay wrt the step size $\boldsymbol{\delta}$
- Perturbing the central path when needed
- Usage of non-local convergence arguments

Can this lead to breaking the $\Omega(m^{1/2})$ barrier for all LPs?

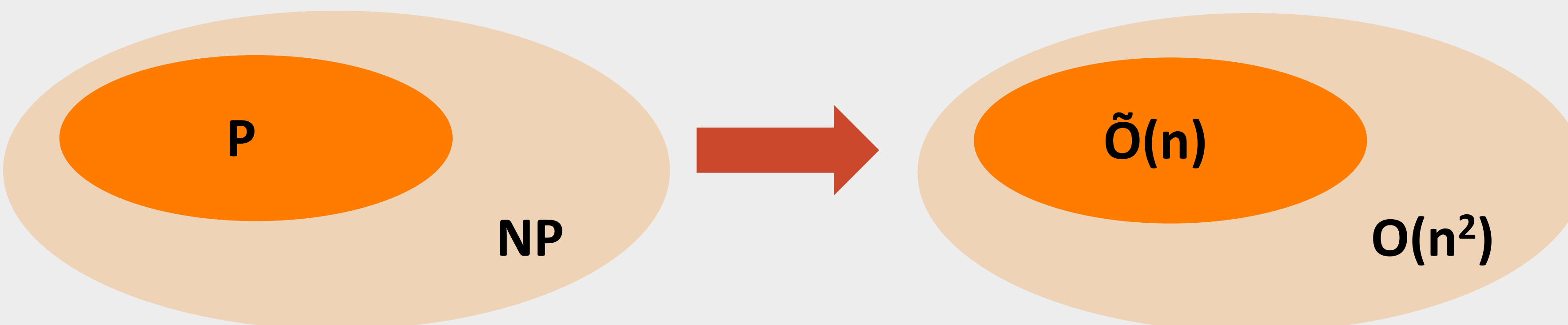
[Lee Sidford '14]: $\tilde{O}(\mathrm{rank}(A)^{1/2})$ iteration bound

Unifying the landscape of algorithmic graph theory

Research on graph algorithms is quite fragmented

Can we establish a more unified picture of the field?

Complexity theory for **sub-quadratic** time algorithms?



Which graph problems can be **efficiently** reduced to other problems?
Study of “max flow”-hardness/completeness?

Bridging the Combinatorial and the Continuous

paths, trees, partitions, routings, matchings, data structures…

↔

matrices, eigenvalues, linear systems, gradients, convex sets…

Powerful approach: Exploiting the interplay of the two worlds

Some other early "success stories" of this approach:

- Spectral graph theory aka the "eigenvalue connection"
- Fast SDD/Laplacian system solvers
- Graph sparsification, random spanning tree generation

...and this is just the beginning!

Thank you

Questions?